Aim: To create a database using array of structures and perform the operations]

1. Add record 2. Display database 3. Search record (Binary search) 4. Modify record

  5. Delete record 6. Sort record (Bubble sort)

Theory:

In this an array of structures is to be created. Parameters such as name, roll number and percentage of student have been used. Roll number is unique.

Add Record: We need to add a new record at the end of database by accepting input from the user.

Display Record: To display the complete database we need to traverse the database using its roll number as key.

Search Record: Here using linear search we have to return the index at which a given roll number is present.

Modify Record: To find a record the student's roll number must be accepted from the user. Then using searching we discover the position then change the data with the data given by user.

Delete Record: To delete a required record, the student's roll number must be accepted from the user. Then using searching we discover the position of the record, and delete its content.

Sort Record: Using Selection Sort to sort in ascending order and insertion sort to sort in descending order. While sort the total data must be sorted.

**Binary Search -**The binary search looks for the element in an already sorted array. The array is divided with a mid and judged in which half the element exits and is then located and displayed.

**Bubble Sort –** In the bubble sort the largest element (in the case of descending order) or the smallest element (in case of ascending order) is bubbled to the top and is hence called the bubble sort algorithm.

SOURCE CODE:

/\*

PROGRAM BY ANOM DEVGUN

GR: 21810017

Implemented a student database using array of structures.

Implemented various operations through switch case.

\*/

#include<iostream>

#include<string.h>

using namespace std;

int nrec;

struct stu

{

int gno;

string fna,lna;

string addr;

};

int main()

{

int rn=0;

struct stu s[100];

int l;

int la=1;

string fn,ln;

string ad;

int i=0,sr,fl=0,j;

int ch;

cout<<"Enter the number of records you want initially\n";

cin>>nrec;

int lb,ub,mid,hl,pa,sw=1;

cout<<"Enter details for each student\n";

for(i=0;i<nrec;i++)

{

cout<<"Enter GR number\n";

cin>>s[i].gno;

cout<<"\nEnter First name of student\n";

cin>>s[i].fna;

cout<<"\nEnter Last name of student\n";

cin>>s[i].lna;

cout<<"\nEnter Address\n";

cin.ignore();

getline(cin,s[i].addr);

}

while(1)

{

cout<<"1)To BubbleSort database\n";

cout<<"2)To display database\n";

cout<<"3)To Binary search record from database\n";

cout<<"4)To delete record from database\n";

cout<<"5)To modify record in database\n";

cout<<"6)To add record to database\n";

cout<<"7)To exit\n";

cin>>ch;

switch(ch)

{

case 1:

for(pa=0;pa<nrec-1&& sw==1;pa++) //BUBBLESORT ARRAY

{

sw==0;

for(j=0;j<nrec-pa-1;j++)

{

if(s[j].gno > s[j+1].gno)

{

sw=1;

hl=s[j].gno;

s[j].gno=s[j+1].gno;

s[j+1].gno=hl;

fn=s[j].fna;

s[j].fna=s[j+1].fna;

s[j+1].fna=fn;

ln=s[j].lna;

s[j].lna=s[j+1].lna;

s[j+1].lna=ln;

ad=s[j].addr;

s[j].addr=s[j+1].addr;

s[j+1].addr=ad;

}

}

}

break;

case 2: //DISPLAY DATABASE

for(i=0;i<nrec;i++)

{

cout<<"GR Number is :"<<s[i].gno<<"\n";

cout<<"Name is "<<s[i].fna<<" "<<s[i].lna<<"\n";

cout<<"Address is : "<<s[i].addr<<"\n";

}

break;

case 3: //BINARY SEARCH

lb=0;

ub=nrec-1;

mid=(lb+ub)/2;

cout<<"Enter Record to be searched by GR number:\n";

cin>>sr;

while(lb <= ub)

{

if(sr == s[mid].gno)

{

cout<<"The record "<<s[mid].gno<<" has been found : \n";

cout<<"Name is "<<s[mid].fna<<" "<<s[mid].lna<<"\n";

cout<<"Address is : "<<s[mid].addr<<"\n";

break;

}

else if(sr > s[mid].gno)

{

lb=mid+1;

}

else

{

ub=mid-1;

}

mid=(lb+ub)/2;

}

if(lb>ub)

{

cout<<"The Record to be searched in the database was not found.\n";

}

break;

case 4 : //TO DELETE RECORD

cout<<"Enter The record you want deleted from database:\n";

cin>>sr;

for(i=0;i<nrec;i++)

{

if(s[i].gno == sr)

{

la=0;

for(j=i;j<nrec;j++)

{

s[j].gno=s[j+1].gno;

s[j].fna=s[j+1].fna;

s[j].lna=s[j+1].lna;

s[j].addr=s[j+1].addr;

}

nrec--;

cout<<"Record deleted.\n";

break;

}

}

if(la == 1)

cout<<"Record not found.\n";

break;

case 5: //TO MODIFY RECORD

cout<<"Input record you want to modify: \n";

cin>>sr;

for(i=0;i<nrec;i++)

{

if(s[i].gno == sr)

{

cout<<"Enter First name and last name: \n";

cin>>s[i].fna>>s[i].lna;

cout<<"Enter Address: \n";

cin.ignore();

getline(cin,s[i].addr);

cout<<"\nRecord updated successfully\n";

la=0;

}

}

if(la == 1)\

cout<<"Record not found.\n";

break;

case 6:

cout<<"Enter Number of Records you want to add to database:\n";

cin>>rn;

for(i=nrec;i<nrec+rn;i++)

{

cout<<"Enter GR number\n";

cin>>s[i].gno;

cout<<"\nEnter First name of student\n";

cin>>s[i].fna;

cout<<"\nEnter Last name of student\n";

cin>>s[i].lna;

cout<<"\nEnter Address\n";

cin.ignore();

getline(cin,s[i].addr);

}

nrec+=rn;

break;

case 7: cout<<"\nNow Exiting.\n";

exit(0);

break;

default:

cout<<"Invalid choice.\n";

}

}

return 0;

}